

# Games for width parameters and monotonicity

Isolde Adler

Department of Informatics, University of Bergen, N-5020 Bergen, Norway  
 isolde.adler@ii.uib.no

**Abstract** We introduce a search game for two players played on a *scenario* consisting of a ground set together with a collection of feasible partitions. This general setting allows us to obtain new characterisations of many width parameters such as rank-width and carving-width of graphs, matroid tree-width and GF(4)-rank-width. We show that the *monotone* game variant corresponds to a tree decomposition of the ground set along feasible partitions. Our framework also captures many other decompositions into ‘simple’ subsets of the ground set, such as decompositions into planar subgraphs.

Within our general framework, we take a step towards characterising monotone search games. We exhibit a large class of *monotone* scenarios, i.e. of scenarios where the game and its monotone variant coincide. As a consequence, determining the winner is in NP for these games. This result implies monotonicity for all our search games, that are equivalent to branch-width of a submodular function.

Finally, we include a proof showing that the matroid tree-width of a graphic matroid is not larger than the tree-width of the corresponding graph. This proof is considerably shorter than the original proof and it is purely graph theoretic.

## 1 Introduction

Search games were introduced by Parsons and Petrov in [34,35,36] and since then gained a lot of interest both in computer science and discrete mathematics [4,9,8,17,31,22,12]. In search games on graphs, a fugitive and a set of searchers move on a graph, according to some rules. The searchers’ goal is to capture the fugitive, and the fugitive tries to avoid capture indefinitely. Depending on the rules, different variants of search games arise.

These games have applications in various areas. On one hand, they are used to model a variety of real-life problems such as searching a lost person in a system of caves [34], clearing tunnels that are contaminated with gas [27], and modeling bugs in distributed environments [15]. On the other hand, search games are strongly related to graph structure theory, especially to width parameters, such as tree-width [7,39], path-width [8], cutwidth [29], directed tree-width [23], and many others. They provide a better understanding of the parameters, since a winning strategy for the cops is a witness for the parameter being small, whereas a winning strategy for the robber is a obstruction for a small parameter. A game characterisation of a width parameter helps in finding examples, and in many cases, games allow for a polynomial time approximation for the problem of deciding whether the corresponding parameter is bounded by some fixed integer  $k$ .

Yet not all width parameters have game characterisations. Our general framework allows us to fill a large part of this gap. We introduce a game parameter that is within a factor of 3 of branch-width of a submodular function. In particular, we obtain games equivalent to rank-width [32], carving-width [41], and GF(4)-rank-width and bi-rank-width [24], and we give an exact game characterisation of matroid tree-width [19] and for tree-width of directed graphs as introduced by Reed in [37]. Moreover, we characterise all our game parameters by a parameter defined via a ‘tree decomposition’. We hope that these new characterisations give a deeper insight, especially into the newer notions such as rank-width, GF(4)-rank-width and bi-rank-width, and maybe even help solving Seese’s conjecture [10].

One very important and desirable property of search game is monotonicity. Intuitively, a search game is *monotone*, if, in the case that the searchers can catch the fugitive, the searchers can catch him without having to search a previously searched area again.

If a search game is monotone, this gives us a polynomial space certificate for proving that determining the winner in NP, because we can restrict ourselves to monotone search strategies only. But not all search games are monotone, and although monotonicity is a well-studied property, until now there is no general method for distinguishing monotone games from non-monotone. Actually, some of the most involved techniques in the area of graph searching were developed for showing monotonicity [27,8,40,25,14,30]. Recent developments in this direction contain new results concerning monotonicity of search games on directed graphs and hypergraphs, and here many important questions remain unsolved [23,6,21,26,1,3]. See [13] for a survey. Since monotonicity has attracted so much interest, a natural question arises: Can we characterise monotone search games?

In this paper, we consider this question and give results that provide a step towards its resolution. We introduce a general framework for a variant of search games where the fugitive is visible, and the searchers – in our case there is just one *captain* – try to corner him by building barriers. In our framework, the fugitive is a robber moving on elements of a finite set  $A$ , and the captain has a collection  $\mathcal{P}$  of ‘feasible’ partitions of  $A$ . In each round, the captain chooses a new feasible partition, and rebuilds the barriers accordingly. The robber tries to escape, but his moves are limited by the (partial) barriers that persist during the process of rebuilding. We also introduce a collection  $\mathcal{S}$  of ‘simple’ subsets of  $A$ . These are subsets of  $A$  where catching the robber is trivial. We say that the captain wins, if she manages to corner the robber in a simple subset. We call such a pair  $(\mathcal{P}, \mathcal{S})$  (satisfying some natural properties) a *scenario* on  $A$ . Generalising ideas of Amini et al. [5], we introduce *weakly submodular* scenarios, and we show that the games on weakly submodular scenarios are monotone.

We keep the assumptions on the scenario very weak, in order to shed light on the conditions that imply monotonicity. Moreover, since we can decompose any scenario, we obtain decompositions of graphs into any kind of ‘simple’ subgraphs, such as planar graphs or  $H$ -minor free-graphs. In this way it should also be possible to find applications in future research.

Our framework also yields a game characterising tree-width of graphs (our game parameter is one less than the number of cops necessary to catch the robber in the robber-and-cops game [40] characterising tree-width).

Let us give an intuition of our game for tree-width: When specialising matroid tree-width to graphs, the notion yields a simple equivalent definition [19,20] of graph tree-width: A *tree decomposition* of a graph  $G$  is then merely a tree  $T$ , whose leaves are labeled by the edges of  $G$ . Every internal tree node  $t \in V(T)$  defines a partition  $P_t$  on the edges  $E(G)$ : the partition corresponding to the leaf labels of the connected components of  $T \setminus t$ . The *width* of  $t$  is then the number of vertices of  $G$  on the *boundaries* of

the sets in  $P_t$  (vertices incident with edges from different partition sets). As usual, the width of a tree decomposition is the maximum of the widths of its tree nodes, and the tree-width of  $G$  is the minimum possible width of a tree decomposition of  $G$ . For every integer  $k \geq 0$  we let  $\mathcal{P}_k$  denote the collection of partitions of  $E(G)$  with boundaries of size at most  $k$ . In the corresponding game, the robber moves on edges of  $G$ , and the captain chooses partitions from  $\mathcal{P}_k$ . The captain has to catch the robber by cornering him on an edge.

The paper is organised as follows. We begin by introducing the captain and robber game on scenarios in Sect. 2. We introduce tree decomposition for scenarios in Sect. 3, and we link tree decompositions to monotone winning strategies. We introduce brambles and we show that they provide a strategy for the robber to escape. In Sect. 4 we introduce weakly submodular scenarios and search trees. We prove monotonicity, linking brambles, search trees, tree decompositions and winning strategies together. Sect. 5 introduces branch decompositions for scenarios and shows how they relate to tree decompositions of scenarios. Sect. 6 contains applications to matroid tree-width, to graph tree-width and to branch-width of connectivity functions, yielding monotone games for each of the invariants. As an aside, Sect. 6.3 contains proof showing that the matroid tree-width of a graphic matroid is not larger than the (traditional) tree-width of the corresponding graph. Our proof is much shorter than the original proof, and it is purely graph theoretic, avoiding the geometric argument in the original proof [19]. Finally, we close with a conclusion in Sect. 7.

## 2 Scenarios and Games

For an integer  $n \geq 1$  let  $[n] := \{1, \dots, n\}$ . A *partition* of a set of  $A$  is a set  $P = \{A_1, \dots, A_d\}$ , consisting of pairwise disjoint subsets  $A_i \subseteq A$  such that  $A = A_1 \dot{\cup} \dots \dot{\cup} A_d$ . We allow the sets  $A_i$  to be empty. Let  $P_1 = \{A_1, \dots, A_d\}$  and  $P_2 = \{B_1, \dots, B_\ell\}$  be two partitions of  $A$  into sets  $A_i \subseteq A$  and  $B_j \subseteq A$ , respectively. We say that  $P_1$  is *coarser* than  $P_2$ ,  $P_1 \geq P_2$ , if every set in  $P_1$  is a union of some sets of  $P_2$ , i.e. for all  $i \in [d]$  there exist  $i_1, \dots, i_n \in [\ell]$  such that  $A_i = B_{i_1} \dot{\cup} \dots \dot{\cup} B_{i_n}$  (we also say that  $P_2$  is *finer* than  $P_1$  and write  $P_2 \leq P_1$ ).

The *common coarsening* of the two partitions is the partition  $P_1 \vee P_2$  of  $A$  into subsets  $X \subseteq A$  that can be written as a union of  $A_i$ s as well as a union of  $B_i$ s, i.e.  $X = A_{i_1} \dot{\cup} \dots \dot{\cup} A_{i_n}$  for some  $i_1, \dots, i_n \in [d]$ , and  $X = B_{i_1} \dot{\cup} \dots \dot{\cup} B_{i_m}$  for some  $i_1, \dots, i_m \in [\ell]$ . By  $\text{Part}(A)$  we denote the collection of all partitions of  $A$ . Note that  $(\text{Part}(A), \leq)$  is a lattice. By  $2^A$  we denote the set of all subsets of  $A$ , and for a subset  $X \subseteq A$ , we let  $X^c := A \setminus X$  denote the complement of  $X$  in  $A$ .

**Definition 21** Let  $A$  be a finite set. A scenario on  $A$  is a pair  $(\mathcal{P}, \mathcal{S})$ , where  $\mathcal{P} \subseteq \text{Part}(A)$  and  $\mathcal{S} \subseteq 2^A$  satisfy

- (SC1)  $\mathcal{P}$  is closed under coarser partitions,
- (SC2) If  $X \subseteq S$  for some  $S \in \mathcal{S}$  and there is a partition  $P \in \mathcal{P}$  with  $X \in P$ , then  $X \in \mathcal{S}$ ,
- (SC3) Every set  $S \in \mathcal{S}$  satisfies  $\{S, S^c\} \in \mathcal{P}$ .

Note that  $\mathcal{P} = \emptyset$  implies  $\mathcal{S} = \emptyset$ , and that  $\mathcal{P} \neq \emptyset$  implies  $\{A\} \in \mathcal{P}$ . Intuitively, the set  $\mathcal{P}$  is the set of ‘feasible’ partitions, and  $\mathcal{S}$  contains the ‘simple’ subsets of  $A$  – the subsets that are ‘well understood’. By (SC1), making partition coarser is ‘feasible’. Condition (SC2) says that if  $S$  is a subset of a ‘simple’ set, and if we can border  $X$  with a feasible partition, then  $X$  is ‘simple’ as well. According to the last condition, ‘simple’ subsets should have a ‘feasible’ border.

**Proviso 22** Throughout the whole paper,  $A$  denotes a nonempty, finite set,  $\mathcal{P}$  denotes a set of partitions of  $A$ , and  $\mathcal{S}$  denotes a collection of subsets of  $A$ .

*The captain and robber game* Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ . The *captain and robber game* on  $(\mathcal{P}, \mathcal{S})$  is a two player game, where one player controls the captain and the other player controls the robber. The robber moves on elements of  $A$ , running within certain subsets of  $A$ . The captain, sitting in her office, lets her assistants build barriers in order to limit the way the robber can move. Such a barrier must be ‘feasible’, so her choice is limited to a set  $\mathcal{P}$  of ‘allowed’ barriers. The captain’s goal is to limit the robber to a set  $S \in \mathcal{S}$  (intuitively, the sets in  $\mathcal{S}$  are well-known areas, where it is easy to catch the robber). The robber’s goal is to avoid being cornered in any set of  $\mathcal{S}$ .

More precisely, in the beginning of a play the captain has not blocked anything, i.e. she chooses the trivial partition  $\{A\}$  of  $A$ , and the robber moves to an arbitrary element of  $A$ . If  $\{A\} \notin \mathcal{P}$ , then the robber wins. Otherwise we have  $\{A\} \in \mathcal{P}$ , and hence the partition  $\{A\}$  is an allowed choice and the game continues. Now suppose the game is in position  $(P, r)$ , where  $P \in \mathcal{P}$  is the partition chosen by the captain and the robber stands on  $r \in A$ . Then  $r \in X \in P$  for some set  $X$  in the partition  $P$ . (The set  $X$  is called the *robber space*.) Now the captain chooses a new partition  $P' \in \mathcal{P}$ . The clever robber finds out which partition  $P'$  she chooses. Now let  $Y \in P \vee P'$  be the subset satisfying  $X \subseteq Y$ . While the barriers are moved from  $P$  to  $P'$ , the captain only blocks  $P \vee P'$ , and the robber can move within the borders of  $P \vee P'$  to a (possibly) new element  $r' \in Y$ . Then the new barrier is built and the robber is in the set  $X' \subseteq Y$  with  $r' \in X' \in P'$ . If  $X' \in \mathcal{S}$ , then the captain wins. Otherwise the play continues. The captain wins if in some step of the play she catches the robber within a set of  $\mathcal{S}$ . Otherwise the robber wins.

The captain has a *winning strategy* on  $(\mathcal{P}, \mathcal{S})$  if she can assure capturing the robber independently of the way the robber moves. Winning strategies for the robber are defined analogously.

The *monotone captain and robber game* on  $(\mathcal{P}, \mathcal{S})$  is defined like the captain and robber game on  $(\mathcal{P}, \mathcal{S})$ , with the additional restriction that all the captain’s moves be *monotone*: Suppose that the play is in position  $(P, r)$ , where  $P \in \mathcal{P}$  is the partition chosen by the captain and the robber

stands on  $r \in X \in P$ . Now the captain may only choose a partition  $P' \in \mathcal{P}$  that refines  $X$ , i.e.  $P'$  contains subsets  $X'_1, \dots, X'_n$  such that  $X = X'_1 \cup \dots \cup X'_n$ . Note that this restriction assures that after moving to  $P'$  the robber space either stays the same or decreases.

Winning strategies for the captain and for the robber in the monotone captain and robber game are defined as usual.

**Remark 23** Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ .

If the captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ , then she has a winning strategy in the (non-monotone) captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .  $\square$

Note that if  $\bigcup \mathcal{S} \subsetneq A$ , then the robber can win by always staying on an element  $r \in A \setminus \bigcup \mathcal{S}$ .

**Definition 24** A scenario  $(\mathcal{P}, \mathcal{S})$  is monotone, if the captain has a winning strategy in the captain and robber game on  $(\mathcal{P}, \mathcal{S})$  if and only if the captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .

In Sect. 4 we exhibit classes of monotone scenarios.

### 3 Tree Decompositions and Brambles for Scenarios

Graphs are finite, simple and undirected, unless stated otherwise. For a graph  $G$  we denote the vertex set by  $V(G)$  and the edge set by  $E(G)$ . For a vertex  $u \in V(G)$  we let  $N_G(u) := \{v \in V(G) \mid \{u, v\} \in E(G)\}$  denote the set of *neighbours* of  $u$  in  $G$  (we omit the subscript  $G$  if it is clear from the context). A tree is a nonempty, connected, acyclic graph. (For the basic notions of graph theory see [11]). For a tree  $T$  let  $L(T)$  denote the set of leaves of  $T$ , i.e. the nodes of degree at most one. We call the nodes of  $V(T) \setminus L(T)$  *internal* nodes. For  $t \in V(T)$  let  $T^{-t}$  denote the set of connected components of  $T \setminus t$ . Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ , and let  $\tau: L(T) \rightarrow \mathcal{S}$  be a mapping from the set of leaves of  $T$  to  $\mathcal{S}$ . For an internal node  $t$  of  $T$  we let  $P_t := \{\bigcup \tau(L(T) \cap V(T')) \mid T' \in T^{-t}\}$ .

A *tree decomposition* for a scenario  $(\mathcal{P}, \mathcal{S})$  is a pair  $(T, \tau)$ , where  $T$  is a tree and  $\tau: L(T) \rightarrow \mathcal{S}$  is a mapping from the set of leaves of  $T$  to  $\mathcal{S}$ , such that

- (TD1) the image  $\tau(L(T)) \subseteq \mathcal{S}$  is a partition of  $A$ , and
- (TD2) all internal nodes  $t \in V(T) \setminus L(T)$  satisfy  $P_t \in \mathcal{P}$ .

Note that we do not require the partition  $\tau(L(T))$  to be in  $\mathcal{P}$ : Assume there exists a partition  $P = \{S_1, \dots, S_n\} \in \mathcal{P}$  with  $S_i \in \mathcal{S}$  for all  $i \in [n]$ . Then the  $n$ -star  $T_n$ , i.e. the tree  $T_n$  consisting of one node  $s$  of degree  $n$  and  $n$  leaves  $t_1, \dots, t_n$ , together with the mapping  $\tau(t_i) := S_i$ , is a tree decomposition for  $(\mathcal{P}, \mathcal{S})$ . Note that if  $\bigcup \mathcal{S} \neq A$ , then  $(\mathcal{P}, \mathcal{S})$  has no tree decomposition.

**Theorem 31** Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ .

The pair  $(\mathcal{P}, \mathcal{S})$  has a tree decomposition if and only if the captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .

A proof sketch of Theorem 31 can be found in the appendix.

A *bramble* for  $\mathcal{P}$  is a nonempty collection  $\mathcal{B}$  of nonempty, pairwise intersecting subsets of  $A$ , such that every partition  $P \in \mathcal{P}$  satisfies  $P \cap \mathcal{B} \neq \emptyset$ . A bramble  $\mathcal{B}$  for  $\mathcal{P}$  *avoids*  $\mathcal{S}$ , if  $\mathcal{B} \cap \mathcal{S} = \emptyset$ .

**Lemma 32** Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ .

If  $\mathcal{P}$  has a bramble avoiding  $\mathcal{S}$ , then the robber has a winning strategy in the captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .

*Proof.* The robber can escape: Whenever the captain chooses a partition  $P \in \mathcal{P}$ , the robber moves to the set  $X \in \mathcal{P} \cap \mathcal{B}$ . This is always possible since any two sets in  $\mathcal{B}$  have a nonempty intersection.  $\square$

## 4 Monotonicity of Weakly Submodular Scenarios

In this section we prove monotonicity for a class of scenarios with *weakly submodular* sets of partitions. The proof uses the notion of search trees for scenarios. For our proof of monotonicity we generalise methods of [28] to scenarios (The paper [28] simplifies and slightly generalises the ideas of [5]).

Let  $P = \{X_1, \dots, X_d\}$  be a partition of  $A$  and let  $F \subseteq A$ . For  $i \in [d]$  let  $P_{X_i \rightarrow F}$  be the partition  $P_{X_i \rightarrow F} := \{X_1 \cap F^c, \dots, X_{i-1} \cap F^c, X_i \cup F, X_{i+1} \cap F^c, \dots, X_d \cap F^c\}$ .

The following notion of weak submodularity is crucial to the proof of monotonicity: In Lemma 46, we need to rearrange partitions induced by tree labelings. Weakly submodular sets of partitions allow for the necessary rearrangements.

We say that a set  $\mathcal{P}$  of partitions of  $A$  is *weakly submodular*<sup>1</sup>, if for any pair of partitions  $P, Q \in \mathcal{P}$  and any pair of sets  $X \in P$  and  $Y \in Q$  with  $A \setminus (X \cup Y) \neq \emptyset$  there exists a nonempty set  $F \subseteq A \setminus (X \cup Y)$  such that  $P_{X \rightarrow F} \in \mathcal{P}$ , or  $Q_{Y \rightarrow F} \in \mathcal{P}$ .

**Definition 41** *Let  $A$  be a finite set. A scenario  $(\mathcal{P}, \mathcal{S})$  on  $A$  is weakly submodular, if  $\mathcal{P}$  is weakly submodular.*

The two following propositions present examples of weakly submodular sets of partitions. Let  $G$  be a graph and let  $P = \{X_1, \dots, X_d\}$  be a partition of  $E(G)$ . Define

$$\partial(P) := \{v \in V(G) \mid \exists e_i, e_j \in E(G) \text{ with } v \in e_i \cap e_j, e_i \in X_i, e_j \in X_j, \text{ and } i \neq j\}.$$

For an integer  $k \geq 1$  we let  $\text{Part}_{\text{tw}}^k := \{P \in \text{Part}(E(G)) \mid |\partial(P)| \leq k\}$ . It is straightforward to check that the following holds.

**Proposition 42** *Let  $G$  be a graph and let  $k \geq 1$  be an integer. Then  $\text{Part}_{\text{tw}}^k$  is a weakly submodular set of partitions, and  $\text{Part}_{\text{tw}}^k$  is closed under coarser partitions.*  $\square$

More generally, we consider sets of partitions arising from connectivity functions. Connectivity functions are strongly related with matroids, and they arise in many different contexts (see i.e. [33]). Let  $f$  be an integer valued function  $f: 2^A \rightarrow \mathbb{Z}$ . The function  $f$  is a *connectivity function* on  $A$ , if

- any subset  $X \subseteq A$  satisfies  $f(X) = f(X^c)$  (*symmetry*),
- any two subsets  $X, Y \subseteq A$  satisfy  $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$  (*submodularity*).

**Example 43** *Let  $G$  be a graph. The function  $\delta: 2^{E(G)} \rightarrow \mathbb{Z}$ , given by  $\delta(X) := |\partial(\{X, X^c\})|$  for  $X \subseteq E(G)$ , is a connectivity function on  $E(G)$ .*

For a connectivity function  $f$  and  $k \in \mathbb{Z}$ , we let  $\text{Part}_f^k := \{P \in \text{Part}(A) \mid \sum_{X \in P} f(X) \leq k\}$ .

**Proposition 44** *Let  $f$  be a connectivity function on  $A$  and let  $k$  be an integer. Then  $\text{Part}_f^k$  is a weakly submodular set of partitions, and  $\text{Part}_f^k$  is closed under coarser partitions.*

*Proof.* Submodularity of  $f$  implies that  $\text{Part}_f^k$  is closed under coarser partitions. It is straightforward to check that  $\text{Part}_f^k$  is a weakly submodular (cf. [5, Sect. 6]).  $\square$

A *bidirected tree* is obtained from an undirected tree with at least one edge by replacing every edge by two edges directed in opposite directions. Directed edges are also called *arcs*. *Neighbours* in a bidirected tree are neighbours in the underlying undirected tree. Let  $T$  be a tree and let  $l: E(T) \rightarrow 2^A$ . For an internal node  $t$  of  $T$  with neighbours  $t_1, \dots, t_n$  we let  $\pi_t := \{l(t, t_i) \mid i \in [n]\}$ .

A *search tree* for  $A$  is a pair  $(T, l)$ , where  $T$  is a bidirected tree, and  $l: E(T) \rightarrow 2^A$  is a labeling function such that

<sup>1</sup> This is a translation from the notion of *weakly submodular partition function* of [28] into the context of scenarios.

- (ST1) for every internal node  $t \in V(T) \setminus L(T)$  the set  $\pi_t$  is a partition of  $A$ , and  
(ST2) the two labels of every 2-cycle are disjoint, i.e.  $l(s, t) \cap l(t, s) = \emptyset$  for all  $(s, t) \in E(T)$ .

A 2-cycle  $st$  of  $T$  is *exact*, if  $l(s, t) \cup l(t, s) = A$ . A search tree  $(T, l)$  is *exact*, if all its 2-cycles are exact. The following is proved in [5].

**Fact 45** *In an exact search tree  $(T, l)$  for  $A$ , the labels of the arcs entering the leaves form a partition of  $A$ .*  $\square$

A label of an arc leaving a leaf is called a *leaf label*. Note that in an exact search tree, a leaf label other than  $A$  cannot appear twice.

A search tree  $(T, l)$  for  $A$  is a *search tree for  $\mathcal{P}$* , where in addition all internal nodes of  $T$  satisfy  $\pi_t \in \mathcal{P}$ . We extend this definition to scenarios. A search tree  $(T, l)$  for  $\mathcal{P}$  is a *search tree for  $(\mathcal{P}, \mathcal{S})$* , if, in addition, every  $(s, t) \in E(T)$  with  $t \in L(T)$  satisfies  $l(s, t) \in \mathcal{S}$ . A search tree  $(T, l)$  is *compatible* with a set  $\mathcal{F} \subseteq 2^A$ , if every leaf label contains an element of  $\mathcal{F}$  as a subset. The following Lemma is proved in the appendix.

**Lemma 46** *Let  $A$  be a finite set, let  $(\mathcal{P}, \mathcal{S})$  be a weakly submodular scenario for  $A$ , and let  $\mathcal{F} \subseteq 2^A$ .*

*If  $(\mathcal{P}, \mathcal{S})$  has a search tree compatible with  $\mathcal{F}$  having at least one internal node, then  $(\mathcal{P}, \mathcal{S})$  has an exact search tree compatible with  $\mathcal{F}$ .*

Note that if  $(T, l)$  is an exact search tree for  $(\mathcal{P}, \mathcal{S})$ , then by Fact 45, the labels entering the leaves of  $T$  form a partition of  $A$  into subsets from  $\mathcal{S}$ . This is the link between search trees and tree decompositions. The proof of the following Theorem is given in the appendix.

**Theorem 47** *Let  $A$  be a finite set and let  $(\mathcal{P}, \mathcal{S})$  be a scenario for  $A$ . If the pair  $(\mathcal{P}, \mathcal{S})$  has an exact search tree, then it has a tree decomposition.*

For  $\mathcal{S} \subseteq 2^A$ , and  $\mathcal{S}$ -bias in  $A$  is a nonempty set  $\mathcal{B} \subseteq 2^A$  of nonempty subsets of  $A$  satisfying

- for every  $S \in \mathcal{S}$  there is a set  $X \in \mathcal{B}$  such that  $S \cap X = \emptyset$ ,
- $\mathcal{B} \cap \mathcal{S} = \emptyset$ .

For example, let  $|A| \geq 2$  and let  $\bigcup \mathcal{S} = A$ . Suppose every partition  $P \in \text{Part}(A)$  with  $|P| \leq 2$  satisfies  $P \not\subseteq \mathcal{S}$ . Then  $\{S^c \mid S \in \mathcal{S}\}$  is an  $\mathcal{S}$ -bias in  $A$ . We remark that for  $\mathcal{S}_1 := \{\{a\} \mid a \in A\}$ , an  $\mathcal{S}_1$ -bias is a *bias* as defined in [5]. Moreover, brambles avoiding  $\mathcal{S}_1$  are precisely the *non-principal* brambles from [5].

The following theorem generalises Theorem 4 of [5] and, with it, Theorem 3.4 of [38].

**Theorem 48** *Let  $A$  be a finite set and let  $(\mathcal{P}, \mathcal{S})$  be a weakly submodular scenario for  $A$ , satisfying  $\bigcup \mathcal{S} = A$ .*

*If the set of partitions  $\mathcal{P}$  has no bramble avoiding  $\mathcal{S}$ , then  $(\mathcal{P}, \mathcal{S})$  has an exact search tree.*

*Proof.* If  $\mathcal{P} = \emptyset$ , then  $\mathcal{S} = \emptyset$  and every bramble avoids  $\mathcal{S}$ . Suppose now that  $\mathcal{P} \neq \emptyset$ . If  $A \in \mathcal{S}$ , then there is no bramble avoiding  $\mathcal{S}$  and we obtain an exact search tree for  $(\mathcal{P}, \mathcal{S})$  by taking two nodes  $s, t$  with labels  $l(s, t) = A$  and  $l(t, s) = \emptyset$ . If there is a bipartition  $\{X, X^c\} \in \mathcal{P}$  satisfying  $\{X, X^c\} \subseteq \mathcal{S}$ , then the two node tree with labels  $X$  and  $X^c$  is an exact search tree for  $(\mathcal{P}, \mathcal{S})$ .

For the rest of the proof, assume that  $A \notin \mathcal{S}$  and that all bipartitions  $P \in \mathcal{P}$  satisfy  $P \not\subseteq \mathcal{S}$ . It is easy to check that in this case, the set  $\mathcal{B}_c := \{S^c \mid S \in \mathcal{S}\}$  is an  $\mathcal{S}$ -bias in  $A$ .

*Claim.* For every  $\mathcal{S}$ -bias  $\mathcal{B}$  in  $A$  there is a search tree for  $\mathcal{P}$  compatible with  $\mathcal{B}$ .

*Proof of the Claim.* Towards a contradiction, assume that there is no search tree for  $\mathcal{P}$  compatible with  $\mathcal{B}$ . Choose  $\mathcal{B}$  of maximum cardinality with this property.

First assume that for every partition  $P \in \mathcal{P}$  there exists a set  $X_P \in \mathcal{P} \cap \mathcal{B}$ . Since  $\mathcal{B}$  is an  $\mathcal{S}$ -bias, we have  $X_P \notin \mathcal{S}$ . Since  $\mathcal{P}$  has no bramble avoiding  $\mathcal{S}$ , there must be two sets  $X, Y \in \mathcal{B}$  with  $X \cap Y = \emptyset$ . But then the 2-cycle labeled  $X$  and  $Y$  is a search tree for  $\mathcal{P}$  compatible with  $\mathcal{B}$ , a contradiction.

Secondly, assume there is a partition  $P = \{X_1, \dots, X_n\} \in \mathcal{P}$  such that  $P \cap \mathcal{B} = \emptyset$ .

(1) For every  $i \in [n]$  satisfying  $X_i \notin \mathcal{S}$  there exists a search tree  $(T_i, l_i)$  for  $\mathcal{P}$  that has exactly one leaf label containing  $X_i$  as a subset, and all other leaf labels contain an element of  $\mathcal{B}$  as a subset.

*Proof of (1).* Let  $i \in [n]$  satisfy  $X_i \notin \mathcal{S}$ . Choose a superset  $X'_i \supseteq X_i$  of maximum cardinality such that  $X'_i \notin \mathcal{B}$ . Then  $\mathcal{B} \cup \{X'_i\}$  is an  $\mathcal{S}$ -bias. By maximality of  $\mathcal{B}$ , there is a search tree for  $\mathcal{P}$  compatible with  $\mathcal{B} \cup \{X'_i\}$ , and by Lemma 46 there is an exact search tree  $(T_i, l_i)$  for  $\mathcal{B} \cup \{X'_i\}$ . If  $(T_i, l_i)$  also is a search tree compatible  $\mathcal{B}$  we are done. Otherwise there exists a leaf  $t_i$  with a leaf label containing  $X'_i$  and containing no other element of  $\mathcal{B}$  as a subset. By maximality of  $X'_i$ , the leaf label is exactly  $X'_i$ . Note that  $X'_i \neq A$ , since  $\mathcal{B}$  contains at least one nonempty set which is not contained in  $X'_i$ . Hence by Fact 45 there is exactly one leaf label  $X'_i$ . This proves (1).

With (1) we complete the proof of the claim. For every  $i \in [n]$  with  $X_i \notin \mathcal{S}$  let  $t_i \in L(T_i)$  be the leaf with the label containing  $X_i$  as a subset. We glue the trees  $T_i$  together by identifying all the nodes  $t_i$  into a new node  $t$ . Then the neighbour  $s_i$  of  $t_i$  in  $T_i$  becomes a neighbour of  $t$  in the new tree  $T$ , and we label  $(t, s_i)$  by  $X_i$  and keep all other labels as in  $(T_i, l_i)$ . For every  $j \in [n]$  satisfying  $X_j \in \mathcal{S}$  we add a new node  $t_j$  to  $T$  via a 2-arc  $tt_j$ , labeling  $(t, t_j)$  by  $X_j$  and  $(t_j, t)$  by  $X_j^c$ . It is easy to see that this gives us a search tree for  $\mathcal{P}$  compatible with  $\mathcal{B}$ . This proves the claim.

Now we choose a search tree  $(T, l)$  for  $\mathcal{P}$  compatible with  $\mathcal{B}_c$ , which exists according to the claim. First suppose  $T$  consists of a single edge  $V(T) = \{s, t\}$  with  $X := l(s, t)$  and  $Y := l(t, s)$ . Then there exist sets  $X_0 \in \mathcal{B}$  and  $Y_0 \in \mathcal{B}_c$  with  $X_0 \subseteq X$  and  $Y_0 \subseteq Y$ . By the definition of  $\mathcal{B}_c$  we have  $\{X_0^c, Y_0^c\} \subseteq \mathcal{S}$ , and by (SC3) we have  $\{X_0, X_0^c\} \in \mathcal{P}$ . We now replace  $Y$  by the new label  $X^c$ . Then the 2-cycle is exact, it remains compatible with  $\mathcal{F}$ , and we have  $X^c \in \mathcal{S}$ . We claim that  $X \in \mathcal{S}$ : This follows from  $X \subseteq Y^c \in \mathcal{S}$  and  $\{X, X^c\} \in \mathcal{P}$  using (SC2). Hence we have found an exact search tree for  $(\mathcal{P}, \mathcal{S})$ .

Secondly, suppose  $T$  has at least one internal node. Let  $s \in L(T)$  and let  $t \in V(T)$  be the neighbour of  $s$ . Then  $l(s, t)$  is a leaf label, and hence there is a subset  $X_{st} \subseteq l(s, t)$  with  $X_{st} \in \mathcal{B}_c$ . Define a new labeling of  $T$  by letting  $l'(s, t) := X_{st}$  and  $l'(t, s) := X_{st}^c$  for all arcs  $(s, t)$  where  $s \in L(T)$ , and letting  $l'(e) = l(e)$  for all other arcs of  $T$ . Then the labels of the arcs entering a leaf are in  $\mathcal{B}$ . By Property (SC2), the pair  $(T, l')$  is a search tree for  $(\mathcal{P}, \mathcal{S})$ . Now we apply Lemma 46 and we obtain an exact search tree for  $(\mathcal{P}, \mathcal{S})$ .  $\square$

Note that the condition  $\bigcup \mathcal{S} = A$  is necessary: Otherwise, we can choose an element  $a \in A \setminus \bigcup \mathcal{S}$  and define the bramble  $\mathcal{B} := \{X \subseteq A \mid a \in X, X \notin \mathcal{S}\}$ . Then  $\mathcal{B}$  avoids  $\mathcal{S}$ , but  $(\mathcal{P}, \mathcal{S})$  has no tree decomposition and hence by Theorem 47 it has no search tree.

Combining Theorem 47, Theorem 31, Theorem 48, Lemma 32, and Remark 23 we obtain the following Corollary.

**Corollary 49 (Characterising Tree Decomposable Scenarios)** *Let  $A$  be a finite set and let  $(\mathcal{P}, \mathcal{S})$  be a weakly submodular scenario on  $A$ , satisfying  $\bigcup \mathcal{S} = A$ . Then the following statements are equivalent.*

1. *The pair  $(\mathcal{P}, \mathcal{S})$  has an exact search tree.*
2. *The pair  $(\mathcal{P}, \mathcal{S})$  has a tree decomposition.*
3. *The captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .*
4. *The captain has a winning strategy in the captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .*
5. *The set of partitions  $\mathcal{P}$  has no bramble avoiding  $\mathcal{S}$ .*

**Corollary 410** *Let  $A$  be a finite set. All weakly submodular scenarios  $(\mathcal{P}, \mathcal{S})$  on  $A$  satisfying  $\bigcup \mathcal{S} = A$  are monotone.*

## 5 Branch Decompositions for Scenarios

Branch-width of graphs is closely related to tree-width of graphs [38]. We generalise branch decompositions to our setting, maintaining the close relation in a natural way. Let  $T$  be a tree and

let  $\beta: L(T) \rightarrow A$  be a mapping. For  $e \in E(T)$ , let  $T_1, T_2$  denote the two connected components of  $T - e$ , and let  $P_e$  denote the pair  $P_e := \{\beta(L(T) \cap V(T_1)), \beta(L(T) \cap V(T_2))\}$  of subsets  $A$ .

A tree  $T$  is cubic, if every internal node of  $T$  has degree 3. Let  $(\mathcal{P}, \mathcal{S})$  be a scenario. A *branch decomposition* for  $(\mathcal{P}, \mathcal{S})$  is a pair  $(T, \beta)$  where  $T$  is a cubic tree, and  $\beta: L(T) \rightarrow \mathcal{S}$  is a mapping, such that

**(BD1)** The image  $\beta(L(T)) \subseteq \mathcal{S}$  is a partition of  $A$ , and

**(BD2)** every edge  $e \in E(T)$  satisfies  $P_e \in \mathcal{P}$ .

There is a close link between branch decompositions and tree decompositions. The following theorem is proved in the appendix.

**Theorem 51** *Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ . If  $(\mathcal{P}, \mathcal{S})$  has a tree decomposition, then  $(\mathcal{P}, \mathcal{S})$  has a branch decomposition.*

Let  $\mathcal{P} \subseteq \text{Part}(A)$ . We define the set of partitions  $\mathcal{P}^3 \subseteq \text{Part}(A)$  by  $\mathcal{P}^3 := \{\{X, Y, Z\} \mid \{\{X, X^c\}, \{Y, Y^c\}, \{Z, Z^c\}\} \subseteq \mathcal{P}, X \dot{\cup} Y \dot{\cup} Z = A\}$ .

**Remark 52** *Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ .*

*If  $(T, \beta)$  is a branch decomposition for  $(\mathcal{P}, \mathcal{S})$ , then  $(T, \beta)$  is a tree decomposition for  $(\mathcal{P}^3, \mathcal{S})$ .*

*Proof.* Every branch decomposition  $(T, \beta)$  for  $\mathcal{P}$  is a tree decomposition for  $(\mathcal{P}^3, \mathcal{S})$ .  $\square$

## 6 Applications to Width Parameters

### 6.1 Branch-width of Connectivity Functions

Given a connectivity function  $f$ , we approximate the branch-width of  $f$  by the captain and robber game. In particular, we obtain a game equivalent to rank-width of graphs [32], and games equivalent to GF(4)-rank-width and bi-rank-width of directed graphs [24]. All these scenarios are monotone. Let  $A$  be a nonempty, finite set, let  $f: 2^A \rightarrow \mathbb{Z}$  be a connectivity function and let  $k$  be an integer. Recall that  $\text{Part}_f^k = \{P \in \text{Part}(A) \mid \sum_{X \in P} f(X) \leq k\}$  (cf. Proposition 44).

**Theorem 61** *Let  $A$  be a finite set and let  $f: 2^A \rightarrow \mathbb{Z}$  be a connectivity function, and let  $\mathcal{S} \subseteq \text{Part}(A)$  be closed under subsets (i.e. if  $S' \subseteq S \in \mathcal{S}$ , then  $S' \in \mathcal{S}$ ). Let  $k$  be an integer satisfying  $k \geq \max\{f(S) \mid S \in \mathcal{S}\}$ . Then*

1.  $(\mathcal{P}_f^k, \mathcal{S})$  is a weakly submodular scenario on  $A$ ,
2.  $(\mathcal{P}_f^k, \mathcal{S})$  satisfies Corollary 49 (Characterising tree decomposable scenarios),
3. In particular, the scenario  $(\mathcal{P}_f^k, \mathcal{S})$  is monotone.

*Proof.* 1: By Proposition 44, the set  $\mathcal{P}_f^k$  is closed under coarser partitions, and it is weakly submodular. In particular,  $(\mathcal{P}_f^k, \mathcal{S})$  satisfies (SC1). By the choice of  $k$ , it satisfies (SC3) as well. Since  $\mathcal{S}$  is closed under subsets, it satisfies (SC2). Hence  $(\mathcal{P}_f^k, \mathcal{S})$  is a weakly submodular scenario on  $A$ . Statements 2 and 3 follow from 1, together with Corollary 49.  $\square$

Let us apply Theorem 61 to a graph  $G$  by letting  $A = E(G)$  and  $f := \delta$  (cf. Example 43). Now we choose our favorite class  $\mathcal{C}$  of graphs that is closed under taking subgraphs (planar,  $H$ -minor free, etc.), and let  $\mathcal{S} := \{S \subseteq E(G) \mid G[S] \in \mathcal{C}\}$ . Choosing  $k$  as in the theorem, we obtain a weakly submodular scenario  $(\mathcal{P}_\delta^k, \mathcal{S})$  satisfying Corollary 49 (Characterising Tree Decomposable Scenarios).

For a connectivity function  $f$  on  $A$  and an integer  $k$ , let  $\mathcal{Q}_f := \{\{X, X^c\} \in \text{Part}(A) \mid f(X) \leq k\}$ . Let  $\mathcal{S}_{\text{sing}} := \{\{a\} \mid a \in A\} \cup \{\emptyset\}$ , and assume that  $k \geq \max\{f(\{a\}) \mid a \in A\}$ . Then, by Theorem 61,  $(\mathcal{Q}_f, \mathcal{S}_{\text{sing}})$  is a scenario. We say that  $f$  has *branch-width* at most  $k$ ,  $\text{bw}(f) \leq k$ , if the scenario  $(\mathcal{Q}_f, \mathcal{S}_{\text{sing}})$  has a branch decomposition. This is equivalent to the conventional definition of branch-width (see i.e. [18]). Note that  $\mathcal{Q}_f^k \subseteq \mathcal{P}_f^k$ , and that  $(\mathcal{Q}_f, \mathcal{S}_{\text{sing}})$  has a branch decomposition if and only if  $(\mathcal{P}_f^k, \mathcal{S}_{\text{sing}})$  has a branch decomposition. Analogously, the following definition extends the definition of tree-width of graphs to tree-width of submodular functions.



**Definition 62** Let  $f$  be a connectivity function on  $A$ , and let  $\mathcal{S}_{\text{sing}} := \{\{a\} \mid a \in A\} \cup \{\emptyset\}$ . Let  $k$  be an integer satisfying  $k \geq \max \{f(\{a\}) \mid a \in A\}$ .

We say that  $f$  has tree-width at most  $k$ ,  $\text{tw}(f) \leq k$ , if the scenario  $(\mathcal{P}_f^k, \mathcal{S}_{\text{sing}})$  has a tree decomposition.

**Corollary 63** 1. If  $(\mathcal{P}_f^k, \mathcal{S}_{\text{sing}})$  has a tree decomposition, then  $(\mathcal{P}_f^k, \mathcal{S}_{\text{sing}})$  has a branch decomposition.

2. If  $(\mathcal{P}_f^k, \mathcal{S}_{\text{sing}})$  has a branch decomposition, then  $((\mathcal{P}_f^k)^3, \mathcal{S}_{\text{sing}})$  has a tree decomposition.

3.  $\text{bw}(f) \leq \text{tw}(f) \leq 3 \cdot \text{bw}(f)$ .

*Proof.* 1 and 2 follow from Theorem 51 and Remark 52. 3 follows from 1 and 2, using submodularity of  $f$ .  $\square$

Hence branch-width and tree-width of a submodular function  $f$  are within a factor of three of each other, and the tree-width of  $f$  can be characterised by a monotone game. In particular, this applies to rank-width of graphs, to GF(4)-rank-width and to bi-rank-width of directed graphs.

**Corollary 64** Rank-width and carving-width of graphs, and both GF(4)-rank-width and bi-rank-width of directed graphs have factor 3 approximations by monotone games, that can also be characterised by tree decompositions.

## 6.2 Tree-width of Matroids

Matroid tree-width was introduced by Hlinený and Whittle in [19]. In this section, we present the scenario for the game characterising matroid tree-width. This scenario is monotone. Moreover, we include a short proof showing that the matroid tree-width of a graphic matroid is not larger than the (traditional) tree-width of the corresponding graph.

Throughout this section, let  $M$  be a matroid with nonempty ground set  $E = E(M)$ , and let  $r$  be the rank function of  $M$  (see [33] for an introduction into matroid theory).

A tree decomposition for  $M$  is a pair  $(T, \iota)$  where  $T$  is a tree and  $\iota: E \rightarrow V(T)$  is an arbitrary mapping. For a node  $x \in V(T)$  let  $T_1^x, \dots, T_d^x$  denote the connected components of  $T - x$ , and let  $F_i^x := \iota^{-1}(V(T_i^x))$  (hence  $F_i^x \subseteq E$ ). The node-width of  $x$  is defined by

$$\text{node-w}(x) = \sum_{i=1}^d r(E \setminus F_i^x) - (d-1) \cdot r(M).$$

The width of the decomposition is the maximum width of the nodes of  $T$ , and the smallest width over all tree decompositions of  $M$  is the (matroid) tree-width of  $M$ , denoted by  $\text{mtw}(M)$ . (The width of an empty tree is 0.)

For a better understanding of node-width, we give two equivalent formulations. Let  $\lambda_M: 2^E \rightarrow \mathbb{N}$  with  $\lambda_M(X) = r(X) + r(E - X) - r(M)$ , denote the connectivity function on  $M$ .

**Remark 65** Let  $(T, \iota)$  be a tree decomposition of a matroid  $M$  and let  $x \in V(T)$ . Then

$$\text{node-w}(x) = r(M) - \sum_{i=1}^d [r(M) - r(E - F_i^x)] = r(M) - \sum_{i=1}^d [r(F_i^x) - \lambda_M(F_i^x)].$$

For a set  $F \subseteq E$  the rank defect (cf. [19]) of  $F$  is given by  $r(M) - r(E - F)$ . So for small node width, the second term intuitively says that we want to maximize the rank defect on the branches of  $T - x$ . Similarly, the third term says that we want to maximize the rank of each branch of  $T - x$  using small cuts.

Notice that we do not require  $\iota$  to be surjective. We can actually restrict the image  $\iota(E(M))$  to the leaves of the decomposition tree (we prove this in the appendix):

**Lemma 66** *Let  $M$  be a matroid with  $\text{mtw}(M) \leq k$ . There is a tree decomposition  $(T, \iota)$  for  $M$  of width at most  $k$  satisfying  $\iota(E(M)) \subseteq L(T)$ .*

Let  $P = \{X_1, \dots, X_d\}$  be a partition of  $E := E(M)$ . The *width* of  $P$  is defined as  $w(P) = \sum_{i=1}^d r(E \setminus X_i) - (d-1) \cdot r(M)$ . Let  $(T, \iota)$  be a tree decomposition for a matroid  $M$  where  $\iota(E(M)) \subseteq L(T)$ . For  $x \in V(T)$  of degree  $d$  let  $P^x := \{F_1^x, \dots, F_d^x\} \in \text{Part}(E(M))$ . Then  $\text{node-w}(x) = w(P^x)$ .

For an integer  $k$  we let  $\text{Part}_{\text{mtw}}^k := \{P \in \text{Part}(E(M)) \mid w(P) \leq k\}$ . Let  $\mathcal{S}_{\text{sing}} := \{\{e\} \mid e \in E(M)\} \cup \{\emptyset\}$ .

**Theorem 67** *Let  $M$  be a matroid with nonempty ground set  $E(M)$  and let  $k \geq 1$  be an integer. Then*

1.  $(\text{Part}_{\text{mtw}}^k, \mathcal{S}_{\text{sing}})$  is a weakly submodular scenario on  $E(M)$ ,
2.  $\text{mtw}(M) \leq k$  if and only if the scenario  $(\text{Part}_{\text{mtw}}^k, \mathcal{S}_{\text{sing}})$  has a tree decomposition, and
3.  $(\text{Part}_{\text{mtw}}^k, \mathcal{S}_{\text{sing}})$  satisfies Corollary 49 (Characterising Tree Decomposable Scenarios).
4. In particular, the scenario  $(\text{Part}_{\text{mtw}}^k, \mathcal{S}_{\text{sing}})$  is monotone.

*Proof.* Statement 1 is straightforward to check using the fact that the rank function  $r$  is a connectivity function. Statement 2 follows from Lemma 66 and the definition of  $(\text{Part}_{\text{mtw}}^k, \mathcal{S}_{\text{sing}})$ . Statement 3 follows from Corollary 49, together with 1 and 2.  $\square$

### 6.3 Tree-width of Graphs and Cycle Matroids

In [19,20] it was shown that matroid tree-width of a cycle matroid  $M[G]$  equals the tree-width of the corresponding graph  $G$ , provided the graph has at least one edge.

We give a much shorter proof showing that the matroid tree-width of a cycle matroid is not larger than the tree-width of the corresponding graph. After translating the definition of matroid tree-width to graphs, the proof is purely graph theoretic.

We close the section by exhibiting the scenario for the game characterising graph tree-width (which is equivalent to the cops and robber game of [40]).

Let  $G$  be a graph. A *tree decomposition* of a graph  $G = (V, E)$  is a pair  $(T, B)$ , consisting of a tree  $T$  and a family  $B = (B_t)_{t \in T}$  of subsets of  $V$ , the *pieces* of  $T$ , satisfying:

- For each  $v \in V$  there exists  $t \in T$ , such that  $v \in B_t$ . (The node  $t$  *covers*  $v$ .)
- For each edge  $e \in E$  there exists  $t \in T$ , such that  $e \subseteq B_t$ . (The node  $t$  *covers*  $e$ .)
- For each  $v \in V$  the set  $\{t \in T \mid v \in B_t\}$  is connected in  $T$ .

The *width* of  $(T, B)$  is defined as  $w(T, B) := \max \{|B_t| - 1 \mid t \in T\}$ . The *tree-width* of  $G$  is defined as  $\text{tw}(G) := \min \{w(T, B) \mid (T, B) \text{ is a tree decomposition of } G\}$ .

By  $M[G]$  we denote the corresponding cycle matroid. For  $F \subseteq E(G)$  we use  $G \upharpoonright F$  for denoting the subgraph  $(V(G), F)$  of  $G$ . For  $F \subseteq E(G)$  let  $G - F := G \upharpoonright (E(G) \setminus F)$ . Let  $c(G)$  denote the number of connected components of  $G$ . (Note that singletons play an important role when counting connected components.) A *vertex-free (VF) tree decomposition* [19] of  $G$  is a pair  $(T, \tau)$ , where  $T$  is a tree, and  $\tau: E(G) \rightarrow V(T)$  is a mapping. For a node  $x \in V(T)$  of degree  $d$ , let again  $T_1^x, \dots, T_d^x$  denote the connected components of  $T - x$ , and let  $F_i^x := \tau^{-1}(V(T_i^x))$ . The *vertex free node-width* of  $x$  is defined by  $\text{VF-node-w}(x) := |V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^d c(G - F_i^x)$ . The *width* of the decomposition is the maximum vertex free node-width of the nodes of  $T$ , and the smallest width over all VF tree decompositions of  $G$  is the *VF tree-width* of  $G$ , denoted by  $\text{VF-tw}(G)$ . (The width of an empty tree is 0.) The following fact is not hard to prove (see [19]).

**Fact 68** *Let  $G$  be a graph containing at least one edge. Then  $\text{mtw}(M[G]) = \text{VF-tw}(G)$ .*

The proof of the following theorem is an aside. It is much shorter than the original proof in [19].

**Theorem 69** *Let  $G$  be a graph with at least one edge. Then  $\text{mtw}(M[G]) = \text{VF-tw}(G) \leq \text{tw}(G)$ .*

*Proof.* The equality follows from Fact 68. Towards a proof of the inequality, we may assume that  $G$  is connected. Let  $(T, B)$  be a *small* tree decomposition of  $G$  of width  $k$ , i.e. there are no two nodes  $s, t \in V(T)$ ,  $s \neq t$ , with  $B_t \subseteq B_s$ . For every  $e \in E(G)$  choose a node  $t_e \in E(T)$  with  $e \subseteq B_{t_e}$ . Define  $\tau: E(G) \rightarrow V(T)$  by  $\tau(e) = t_e$ . Note that since  $(T, B)$  is small, the set  $L(T)$  of leaves of  $T$  is contained in the image of  $E(G)$  under  $\tau$ , we have  $L(T) \subseteq \tau(E(G))$ .

Let  $x \in V(T)$ ,  $d = \deg(x)$ . We show that the node-width of  $x$  is at most  $|B_x| - 1$ . Towards this, let  $\partial F_i^x$  denote the set of vertices incident with an edge in  $F_i^x$  and an edge in  $E(G) \setminus F_i^x$ . From  $L(T) \subseteq \tau(E(G))$  it follows that  $E(G - F_i^x) \neq \emptyset$  and hence  $1 + |V(F_i^x) \setminus \partial F_i^x| \leq c(G - F_i^x)$ , for  $i = 1, \dots, d$ . Moreover, (TD3) implies that  $\bigcup_{i=1}^d \partial F_i^x \subseteq B_x$ . The node-width of  $x$  is

$$\begin{aligned} |V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^d c(G - F_i^x) &= |V(G)| + (d-1) \cdot 1 - \sum_{i=1}^d c(G - F_i^x) \\ &\leq |V(G)| + d - 1 - \sum_{i=1}^d (1 + |V(F_i^x) \setminus \partial F_i^x|) = |V(G)| - 1 - \sum_{i=1}^d |V(F_i^x) \setminus \partial F_i^x| = -1 |B_x| - 1. \end{aligned}$$

The last equality follows from  $V(G) = \bigcup_{i=1}^d (V(F_i^x) \setminus \partial F_i^x) \dot{\cup} B_x$ .  $\square$

As mentioned above, the inequality can actually be replaced by an equality.

**Fact 610 (Hlinený, Whittle [19,20])** *Any graph  $G$  with at least one edge satisfies  $\text{mtw}(M[G]) = \text{VF-tw}(G) = \text{tw}(G)$ .*

Fact 610 implies that Theorem 67 specialises to graphs (with at least one edge) in the expected way. Nevertheless, let us make the scenario for tree-width of graph explicit. Let  $\mathcal{S}_{\text{sing}} := \{\{e\} \mid e \in E(G)\} \cup \{\emptyset\}$ . Recall that for an integer  $k \geq 1$ ,  $\text{Part}_{\text{tw}}^k = \{P \in \text{Part}(E(G)) \mid |\partial(P)| \leq k\}$ . Using Proposition 42, it is easy to check that the following holds.

**Theorem 611** *Let  $G$  be a graph containing at least one edge and let  $k > 1$  be an integer. Then*

1.  $(\text{Part}_{\text{tw}}^k, \mathcal{S}_{\text{sing}})$  is a weakly submodular scenario on  $E(G)$ ,
2.  $\text{tw}(G) \leq k$  if and only if the scenario  $(\text{Part}_{\text{tw}}^k, \mathcal{S}_{\text{sing}})$  has a tree decomposition, and
3.  $(\text{Part}_{\text{tw}}^k, \mathcal{S}_{\text{sing}})$  satisfies Corollary 49 (Characterising Tree Decomposable Scenarios).

## 7 Conclusion

We introduced *scenarios* and the *captain and robber game* played on a scenario. We proved that in all games on *weakly submodular* scenarios, the captain has a winning strategy, if and only if the captain has a *monotone* winning strategy, i.e. the games on weak submodularity scenarios are *monotone*. Extending ideas of [5], the proof uses search trees, tree decompositions of scenarios, and brambles in scenarios.

Our result implies monotonicity for a class of search games, that are equivalent to branch-width of submodular functions. We obtain an exact characterisation for matroid tree-width by a monotone game, we obtain a monotone game equivalent to rank-width of graphs, and a monotone game characterising tree-width. Beyond this, our framework also captures decompositions into ‘simple’ subsets of the ground set.

We also included a proof showing that the matroid tree-width of a graphic matroid is not larger than the tree-width of the corresponding graph. This proof is much shorter than the original proof [19] and purely graph theoretic.

Moreover, with our framework it is easy to define a notion of *branch-width* for directed graphs, that is within a factor of 3 of the tree-width of a directed graph, as introduced in [37]. We also obtain an exact game characterisation for tree-width of directed graphs. These various applications give reason to believe that our framework may also be useful in future research – providing a tool for producing game characterisations, that even come with a notion of decomposition.

For scenarios that are not weakly submodular, it is still open whether monotonicity holds. The games characterising hypertree-width [16] and directed tree-width [23] are not monotone. Nevertheless, the monotone and the non-monotone variants are strongly related [23,3]. In both cases, this relation is obtained via the notion of *separators*. Can we obtain similar results for scenarios not satisfying weak submodularity? Can we extend the definition of scenarios and obtain more insight into the open problems concerning monotonicity of the games for DAG-width and Kelly-width [26]?

The author thanks Tomáš Gavenčíak and Marc Thurley for comments on drafts of this paper.

## References

1. I. Adler. Marshals, monotone marshals, and hypertree-width. *Journal of Graph Theory*, 47:275–296, 2004.
2. I. Adler. Tree-related widths of graphs and hypergraphs. *SIAM J. Discrete Math.*, 22(1):102–123, 2008.
3. I. Adler, G. Gottlob, and M. Grohe. Hypertree width and related hypergraph invariants. *Eur. J. Comb.*, 28(8):2167–2181, 2007.
4. M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Appl. Math.*, 8:1–11, 1984.
5. O. Amini, F. Mazoit, N. Nisse, and S. Thomassé. Submodular partition functions. *J. Discrete Mathematics*, 2008. accepted.
6. D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. Dag-width and parity games. In B. Durand and W. Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 524–536. Springer, 2006.
7. D. Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science*, 5:33–49, 1991.
8. D. Bienstock and P. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.
9. H. Bodlaender and D. Thilikos. Computing small search numbers in linear time. In R. Downey, M. Fellows, and F. Dehne, editors, *IWPEC*, volume 3162 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2004.
10. B. Courcelle and S.I. Oum. Vertex-minors, monadic second-order logic, and a conjecture by seese. *J. Comb. Theory, Ser. B*, 97(1):91–126, 2007.
11. R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.
12. F. Fomin, P. Golovach, and J. Kratochvíl. On tractability of cops and robbers game. In Giorgio Ausiello, Juhani Karhumäki, Giancarlo Mauri, and C.-H. Luke Ong, editors, *IFIP TCS*, volume 273 of *IFIP*, pages 171–185. Springer, 2008.
13. F. Fomin and D. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.
14. P. Fraigniaud and N. Nisse. Monotony properties of connected visible graph searching. *Inf. Comput.*, 206(12):1383–1393, 2008.
15. M. Franklin, Z. Galil, and M. Yung. Eavesdropping games: a graph-theoretic approach to privacy in distributed systems. *J. ACM*, 47(2):225–243, 2000.
16. G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: Game theoretic and logical characterizations of hypertree width. *Journal of Computer and System Sciences*, 66:775–808, 2003.
17. M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 289–298, New York, NY, USA, 2006. ACM.
18. P. Hliněný and S.I. Oum. Finding branch-decompositions and rank-decompositions. In L. Arge, M. Hoffmann, and E. Welzl, editors, *ESA*, volume 4698 of *Lecture Notes in Computer Science*, pages 163–174. Springer, 2007.
19. P. Hliněný and G. Whittle. Matroid tree-width. *Eur. J. Comb.*, 27(7):1117–1128, 2006.
20. P. Hliněný and G. Whittle. Addendum to matroid tree-width. *Eur. J. Comb.*, 30(4):1036–1044, 2009.
21. P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. In N. Bansal, K. Pruhs, and C. Stein, editors, *SODA*, pages 637–644. SIAM, 2007.
22. P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008.
23. T. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *Journal of Combinatorial Theory(Series B)*, 82:128–154, 2001.

24. M.M. Kanté. The rank-width of directed graphs. *CoRR*, abs/0709.1433, 2007.
25. L.M. Kirousis and C.H. Papadimitriou. Searching and pebbling. *Theor. Comput. Sci.*, 47(3):205–218, 1986.
26. S. Kreutzer and S. Ordyniak. Digraph decompositions and monotonicity in digraph searching. In H. Broersma, T. Erlebach, T. Friedetzky, and D. Paulusma, editors, *WG*, volume 5344 of *Lecture Notes in Computer Science*, pages 336–347, 2008.
27. A. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40(2):224–245, 1993.
28. L. Lyaudet, F. Mazoit, and S. Thomassé. Partitions versus sets: a case of duality. *CoRR*, abs/0903.2100, 2009.
29. F. Makedon and I.H. Sudborough. On minimizing width in linear layouts. *Discrete Applied Mathematics*, 23(3):243–265, 1989.
30. F. Mazoit and N. Nisse. Monotonicity of non-deterministic graph searching. In A. Brandstädt, D. Kratsch, and H. Müller, editors, *WG*, volume 4769 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2007.
31. J. Obdržálek. Dag-width: connectivity measure for directed graphs. In *SODA*, pages 814–821. ACM Press, 2006.
32. S.I. Oum and P.D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006.
33. J.G. Oxley. *Matroid Theory*. Oxford University Press, 1992.
34. T.D. Parsons. Pursuit evasion in a graph. In *Theory and Application of Graphs*, volume 642 of *Lecture Notes in Math.*, pages 426–441. Springer Verlag, 1976.
35. T.D. Parsons. The search number of a connected graph. In *Proceedings of the 9th Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume XXI of *Congress. Numer., Utilitas Math.*, pages 549–554. 1978.
36. N.N. Petrov. A problem of pursuit in the absence of information on the pursued. In *Differentsial'nye Uravneniya*, volume 18, pages 1345–1352. 1982.
37. B. Reed. Introducing directed tree width. In *6th Twente Workshop on Graphs and Combinatorial Optimization, Electron. Notes Discrete Math.* 3, Enschede, Holland, 1999.
38. N. Robertson and P.D. Seymour. Graph minors X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52:153–190, 1991.
39. P.D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory, Ser. B*, 58(1):22–33, 1993.
40. P.D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58:22–33, 1993.
41. P.D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

## A Appendix

We restate and prove Theorem 31:

**Theorem A1** *Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ .*

*The pair  $(\mathcal{P}, \mathcal{S})$  has a tree decomposition if and only if the captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ .*

We only sketch the proof. For graphs and hypergraphs, a detailed proof of similar spirit can be found in [2, Sect. 4].

*Proof sketch.* Let  $(T, \tau)$  be a tree decomposition for  $(\mathcal{P}, \mathcal{S})$ . If  $T$  has no internal nodes, then it is easy to see that the captain can win by using the partition given by  $\tau$ . Otherwise, the captain chooses an internal node  $t$  and moves to  $P_t$ . Then the robber chooses a set  $X \in P_t$ . This set  $X$  corresponds to the labels of the leaves of exactly one component  $T'$  of  $T^{-t}$ . The captain then chooses the neighbour  $s$  of  $t$  in  $T'$  and moves to  $P_s$ . In this way the captain finally catches the robber in a leaf of  $T$ .

Conversely, suppose the captain has a winning strategy in the monotone captain and robber game on  $(\mathcal{P}, \mathcal{S})$ . Then the captain's strategy tree gives rise to a tree decomposition of  $(\mathcal{P}, \mathcal{S})$ . (The nodes of the strategy tree correspond to the captain's partitions. The strategy tree has the captain's first partition as a root, and a partition  $\mathcal{P}$  has a successor for every set  $X \in \mathcal{P}$  that the robber can reach while the captain moves to  $\mathcal{P}$ .)  $\square$

We restate and prove Lemma 46:

**Lemma A2** *Let  $A$  be a finite set, let  $(\mathcal{P}, \mathcal{S})$  be a weakly submodular scenario for  $A$ , and let  $\mathcal{F} \subseteq 2^A$ .*

*If  $(\mathcal{P}, \mathcal{S})$  has a search tree compatible with  $\mathcal{F}$  having at least one internal node, then  $(\mathcal{P}, \mathcal{S})$  has an exact search tree compatible with  $\mathcal{F}$ .*

*Proof.* Let  $(T, l)$  be a search tree with at least one internal node for  $(\mathcal{P}, \mathcal{S})$ , that is compatible with  $\mathcal{F}$ . We choose  $l$  amongst all possible labelings such that the sum

$$\sum_{t \in V(T) \setminus L(T)} \sum_{X \in \pi_t} |X| + \sum_{\substack{s \in L(T) \\ s' \in N(s)}} |l(s, s')| \quad (1)$$

is maximal. Suppose  $st$  is a 2-cycle of  $T$  that is not exact.

If, say,  $s$  is a leaf, then we can replace  $l(s, t)$  by  $l(t, s)^c$ . If neither of  $s$  and  $t$  is a leaf, then, by maximality of Sum (1), for every nonempty set  $F \subseteq A \setminus (l(s, t) \cup l(t, s)^c)$  (such a set  $F$  exists!) we have  $(\pi_s)_{l(s, t) \rightarrow F} \notin \mathcal{P}$ . Hence, since  $\mathcal{P}$  is weakly submodular, we can replace  $\pi_t$  by  $(\pi_t)_{l(t, s) \rightarrow F}$ . In both cases we obtain a search tree  $(T, l')$  for  $(\mathcal{P}, \mathcal{S})$  compatible with  $\mathcal{F}$ , where the size of Sum (1) is strictly increased, a contradiction.  $\square$

We restate and prove Theorem 47:

**Theorem A3** *Let  $A$  be a finite set and let  $(\mathcal{P}, \mathcal{S})$  be a scenario for  $A$ . If the pair  $(\mathcal{P}, \mathcal{S})$  has an exact search tree, then it has a tree decomposition.*

*Proof.* We show that if the pair  $(\mathcal{P}, \mathcal{S})$  has an exact search tree  $(T, l)$ , then we obtain a tree decomposition  $(T, \tau)$  for  $(\mathcal{P}, \mathcal{S})$  by letting  $\tau(t) := l(s, t)$  for  $t \in L(T')$ . Since  $(T, l)$  is a search tree for  $(\mathcal{P}, \mathcal{S})$ , the mapping  $\tau$  is indeed a mapping from  $L(T)$  to  $\mathcal{S}$ . By Fact 45, (TD1) is satisfied. If  $T$  has at most one internal node, (TD2) is obviously satisfied as well.

For a 2-arc  $st$  let  $T_t$  denote the subtree of  $T$  obtained by removing the arcs  $(s, t)$  and  $(t, s)$  from  $T$ , that contains  $t$ . The following claim implies that Condition (TD2) holds.

*Claim.* All 2-cycle  $st$  with two internal nodes  $s$  and  $t$  satisfy

$$l(s, t) = \bigcup_{\substack{v \in L(T) \cap V(T_t) \\ u \in N(v)}} l(u, v).$$

Towards proving the claim, let  $T'_t$  be obtained from  $T_t$  by adding vertex  $s$  and the two arcs  $(t, s)$  and  $(s, t)$ . Then  $(T'_t, l \upharpoonright V(T'_t))$  is an exact search tree for  $A$ , and hence by Fact 45, the labels of the arcs entering the leaves of  $T'_t$  form a partition of  $A$  and the claim follows.  $\square$

We restate and prove Theorem 51:

**Theorem A4** *Let  $(\mathcal{P}, \mathcal{S})$  be a scenario on  $A$ . If  $(\mathcal{P}, \mathcal{S})$  has a tree decomposition, then  $(\mathcal{P}, \mathcal{S})$  has a branch decomposition.*

*Proof.* Let  $(T, \tau)$  be a tree decomposition for  $(\mathcal{P}, \mathcal{S})$ . We turn  $T$  into a cubic tree  $T'$  by replacing every internal node  $t$  of  $T$ , that has at least four neighbours  $t_1, \dots, t_n$  ( $n \geq 4$ ), by a cubic tree with leaves  $t_1, \dots, t_n$ . Identifying  $L(T')$  with  $L(T)$  in the obvious way, and using the fact that  $\mathcal{P}$  is closed under coarser partitions, it is easy to see that  $(T', \tau)$  is a branch decomposition for  $(\mathcal{P}, \mathcal{S})$ .  $\square$

We restate and prove Lemma 66:

**Lemma A5** *Let  $M$  be a matroid with  $\text{mtw}(M) \leq k$ . There is a tree decomposition  $(T, \iota)$  for  $M$  of width at most  $k$  satisfying  $\iota(E(M)) \subseteq L(T)$ .*

*Proof.* Let  $(T', \iota')$  be a tree decomposition for  $M$  of width at most  $k$ . For every element  $e \in E(M)$  satisfying  $\iota'(e) = t$ , where  $t \in V(T')$  is an internal node, we create a new neighbour  $t_e$  of  $t$  and let  $\iota(e) := t_e$ . It is easy to verify that in this way we obtain the desired tree decomposition  $(T, \iota)$  for  $M$ .  $\square$